(1)

Virginia Polytechnic Institute and State University

Computer Science
Industrial Engineering and Operations Research
BLACKSBURG, VIRGINIA 24061

CSIE-83-16 January 1984

HUMAN-COMPUTER INTERACTIONS AND DECISION BEHAVIOR

FINAL REPORT

Robert C. Williges
Roger W. Ehrich
Beverly H. Williges
H. Rex Hartson
Joel S. Greenstein
John W. Roach
Timothy E. Lindquist

Prepared for
Engineering Psychology Programs, Office of Naval Research
ONR Contract Number N00014-81-K-0143
Work Unit NR SRO-101

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: CSIE-83-16

2. GOVT ACCESSION NO.: AD-A139759

3. RECIPIENT'S CATALOG NUMBER:

4. TITLE (and Subtitle): HUMAN-COMPUTER INTERACTIONS AND DECISION BEHAVIOR

5. TYPE OF REPORT & PERIOD COVERED: Final Report

6. PERFORMING ORG. REPORT NUMBER:

7. AUTHOR(s): Robert C. Williges, Roger W. Ehrich, Beverly H. Williges, H. Rex Hartson, Joel S. Greenstein, John W. Roach, Timothy E. Lindquist

8. CONTRACT OR GRANT NUMBER(s): N00014-81-K-0143

9. PERFORMING ORGANIZATION NAME AND ADDRESS: Industrial Engineering & Operations Research, Virginia Polytechnic Institute & State University, Blacksburg, VA 24061

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 61153N42; RR04209; RR0420901; NR SRO-101

11. CONTROLLING OFFICE NAME AND ADDRESS: Office of Naval Research, Code 442, 800 North Quincy Street, Arlington, VA 22217

12. REPORT DATE: January 1984

13. NUMBER OF PAGES: 58

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office):

15. SECURITY CLASS. (of this report): Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

human-computer systems, human-computer dialogues, human-computer interface, human-computer communications, dialogue management system, dialogue author, programmer, end user, software interface

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This is the final report of a three year research program directed toward understanding and improving human-computer dialogue. This report provides an overview of the research program, the scientific personnel who worked on the project, the major scientific accomplishments, the list of reports and archival publications, the technical information exchange, the interdisciplinary graduate training and the project impact for future research.

PREFACE

This research was supported by the Office of Naval Research under ONR Contract Number N00014-81-K-0143, Work Unit Number NR SRO-101. The effort was supported by the Engineering Psychology Programs, Office of Naval Research, under the technical direction of Dr. John J. O'Hare. Representatives from the following Navy laboratories served on the Scientific Advisory Panel for this research.

Naval Air Development Center
Naval Research Laboratory
Naval Personnel Research and Development Center
Naval Surface Weapons Center
Naval Training Equipment Center
Naval Underwater Systems Center
Pacific Missile Test Center

TABLE OF CONTENTS

RESEARCH PROGRAM

Overview

For many years now, the computer boom has created an unparalleled demand for sophisticated software systems, and, since its onset, the demand for functionality has been so large that much of the design effort has been concentrated on the system code. With the steady increase in software sophistication has come the realization that the state of designs for human-computer interfaces is lagging behind, and computing system users are beginning to demand better treatment. Management of human-computer dialogues is essential for the enhancement of the information processing and decision making capabilities of computer users working in real-time, demanding environments.

A solution to the interface problem requires a comprehensive approach that considers the applications programmer, the dialogue author, and the end user as well as the environments in which they must function. Our research has dealt not only on specific dialogue techniques, but also on the nature of software design methodologies, the tools necessary to facilitate the development of high-quality, flexible human-computer interfaces, and the allocation of tasks between the human and the computer.

This SRO program involved a multidisciplinary team of scientists from the Department of Computer Science (CS) and the Department of Industrial Engineering and Operations Research (IEOR) conducting programmatic research to develop principles and techniques for effective human-computer communication. Expertise within the research team included programming languages, computer graphics, database systems, artificial intelligence, modeling, training, and human factors.

A sketch of the major aspects of the program is shown in Figure 1. Essentially, there were four major thrust areas of research which had the combined goal of understanding the factors that might lead to improved human-computer communication through the development of quality software interfaces. The major components of the research program included the Dialogue Management System (DMS) operating environment, the role of the dialogue author, the role of the applications programmer, and the end user interface.

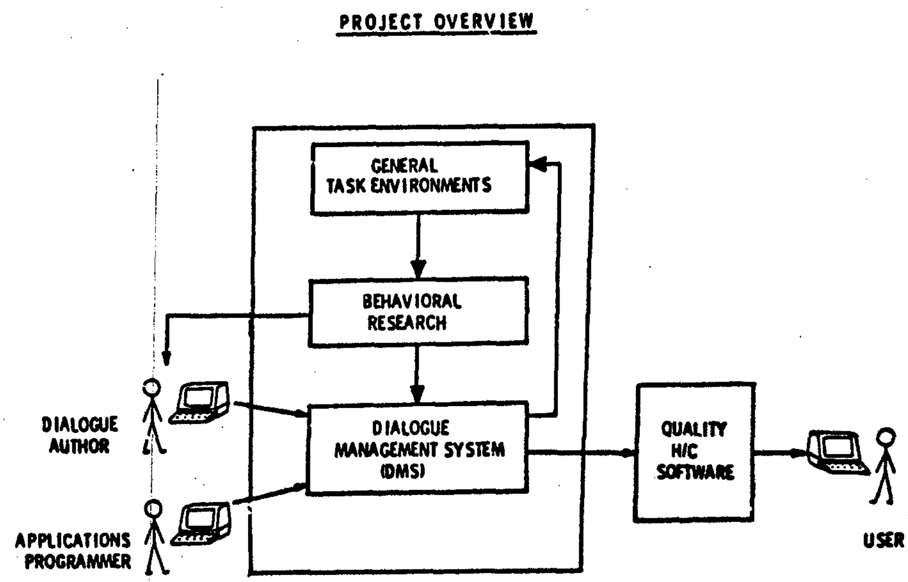

Figure 1. *Program overview.*

DMS methodology and operating environment. The core of the research is the software system, the DMS, which is a complete system for defining, modifying, simulating, executing, and metering human-computer dialogues. Research emphasis in the DMS system involved the development of an execution environment that supports a Generic Environment for Interactive Experiments (GENIE) as a means of conducting behavioral research; the construction of an overall human-computer system development methodology which provides the developer with a graphical programming language for producing an executable software requirements specification; and new methods for modeling, designing, and developing human-computer interfaces based on syntactic and semantic specification. The DMS environment is based upon the hypothesis that dialogue software should be designed separately from the code that implements the computational parts of an application. To achieve the goals of DMS two distinct roles were defined for the production of a software application, the dialogue author and the applications programmer.

Dialogue author. The role of the dialogue author is an extremely important concept that has evolved during this research effort. If the dialogue, or communication, component of a software system can be kept independent of the computational component of the system, one can develop, modify, and monitor the communication interface in a straightforward and efficient manner. Much of the research conducted on the dialogue author was concerned with the development of tools to facilitate the authoring process which, in turn, is based on a transaction model. The toolkit developed for authoring dialogues involves a variety of components including a dynamic language executor, a behavioral demonstrator to exercise prototype interfaces at various stages of design or implementation, and a dialogue author interface called AIDE (Author's Interactive Design Environment). Dialogue authors, for example, may use these interface design tools to implement keypad, menu, textual, and graphic designs.

Applications programmer. Research dealing with the programmer's role dealt primarily with techniques and procedures that enabled the programmers and dialogue authors to work separately but cooperatively under the DMS environment, levels of interactiveness in programming environments, and the application of software metrics to human-computer interfaces. In addition to these activities, a corollary effort was directed toward validation of the ADA programming support environments.

End user. A third role shown in Figure 1 is the end user. Research dealing with the end user was divided into three major areas. The first was in the area of sophisticated dialogue design for novice users, and here some of the problems investigated include command language design, adaptive information retrieval, interactive help information, and expert design systems. The second area concerned new technologies, specifically voice input and output. Here problems of interest included format of synthesized emergency messages and user feedback and error correction with voice recognition. The third area dealt with dynamic human-computer task allocation. Both dialogue-based communication and model-based communication for dynamic task allocation were studied.

Project Team

Project directors. Roger W. Ehrich and Robert C. Williges served as co-principal investigators. Seven research faculty directed the research program. Of these researchers only one had received support from the Office of Naval Research prior to this contract. The research was divided into five subareas. The subareas and the faculty members responsible for each are given below.

Human-Computer Dialogue Management System
(Roger W. Ehrich, H. Rex Hartson, John W. Roach)

Software Human Engineering
(Timothy E. Lindquist)

Interactive Computer Display Principles
(Robert C. Williges)

Voice Input/Output Management
(Beverly H. Williges)

Human-Computer Task Allocation
(Joel S. Greenstein)

Research personnel. These seven faculty directors, as well as other supporting research faculty, directed 22 different graduate students during the contract period. In addition, 6 individuals served as programmers for the project during various stages of the research. Individuals who participated in the project included:

Faculty

Roger W. Ehrich
Robert D. Foley
Joel S. Greenstein
Robert M. Haralick
H. Rex Hartson
Gerald F. Kehoe
Timothy E. Lindquist
John W. Roach
Beverly H. Williges
Robert C. Williges

Graduate Students

S. Ahmed
V. Batra
B. Chao
A. Cohill
J. Elkerton
R. Fainter
L. Folley
M. Hakkinen
D. Johnson
C. Ku
S. Lam
J. Lee
L. Matthews
P. Narang
J. Pittman
M. Revesman
C. Rieger
J. Schurick
A. Siochi
T. Spine
M. Stinson
T. Yunten

Programmers

J. Brandenburg
J. Callan
R. Critz
R. Fainter
J. Goodwin
J. Maynard

Visiting faculty. Because of the scope of this project, a distinguished European researcher, Professor Brian Shackel, spent two extended periods working with our research team. Professor Shackel is head of the Department of Human Sciences, University of Technology, Loughborough, England, and is well-known for his research on various aspects of the human-computer interface problem. He participated fully in the research program and was involved in several of our intensive one-day workshops for members of the research team.

RESEARCH FINDINGS

Our research program resulted in a large variety of behavioral research studies and software implementations. The emphasis of these activities are summarized in terms of human-computer dialogue models/concepts, dialogue design tools, and dialogue principles. Detailed discussions of each of these areas are provided in the technical reports and archival publications prepared on those projects. Findings for each of these areas are described separately.

Dialogue Models/Concepts

Many of the products resulting from this research program are direct outgrowths of fundamental conceptualization and model developments which were a focal point of the basic research. First, a series of dialogue concepts evolved. These included the concept of dialogue independence, dialogue author, human-human communication model, a transaction model, interface models, supervisory methodology, holistic design, meta-communication model, and cognitive structures. Second, techniques for building user models for commanding computer action and for dynamic task allocation were developed for the end user interface.

Dialogue Independence. Dialogue independence refers to the separation of dialogue from computation in the design and structuring of software systems. Dialogue independence means that issues of human-computer interface design can be treated separately from issues of computational software design and that special tools and techniques can be applied to the design of human-computer interfaces. Dialogue independence is one of the most fundamental principles to be formulated in this project and leads to the structure of DMS and the supervisory methodology.

Dialogue author. One of the goals of the Dialogue Management System is to provide methodology and tools so that a new specialist called the dialogue author can assume the task of designing, implementing, testing, and maintaining human-computer interfaces. Human interfaces should be specified and designed by behavioral specialists who are trained in the field. This contrasts sharply with conventional methodology in which designs are done largely by computer scientists, occasionally with a human factors specialist in a consulting role.

Human-human communication. We originally assumed that human-human communication would work as an ideal model for human-computer communication. Yet, many important elements of human-human communication, age, sex, race, are totally absent from human-computer communication. We find as a consequence that communication patterns with machines are far different from communication patterns between people. Pure information transfer, not communication, is the normal mode of human-computer interaction.

Transaction model. In order to provide a suitable dialogue author's toolkit, a model is required that describes the functional components of human-computer interaction and their interrelationships. The transaction model that has been developed is based upon the hypothesis that the primary issue in human-computer interaction is communication and that the vehicle for achieving communication is language. Accordingly, the transaction model employs a linguistic model consisting of parts that relate computer responses to user inputs and a powerful parser for user inputs.

Interface models. Through an examination of the specification and validation needs of Ada Programming Support Environments (APSE), an interface specification technique has been developed. By elaborating the syntax functionality, protocols, and limits of all components, an interface can be completely specified. The specification technique has been applied to two different areas: software-software interfaces and human-computer interfaces. The Common APSE Interface Set (CAIS) has been used to develop the specification technique. This software-software interface defines the kernel set of facilities available to tools in Ada support environments. Applying the technique to the CAIS has raised issues regarding the current design of the CAIS and regarding the procedures for evaluating different implementations of the CAIS. The technique has also been applied to human-computer interfaces. Here, the interface components that are specified include the commands invoked, the manual actions performed, and the decisions made by the user. The syntax, functionality, protocols and limits of these components are detailed in procedural form allowing analysis by any of a number of software tools. Aside from providing an appropriate analysis technique for human-computer interfaces, this technique has surfaced a dimension of human-computer interactions needing further investigation.

Supervisory methodology. Crucial to DMS is a methodology that provides a mechanism by means of which the specifications of a software system can be organized and the necessary decomposition of computational and dialogue components can be achieved. The supervisory methodology provides a systematic procedure for the development of a software system that eliminates need for reimplementing the system many times during its development. It also provides for the decomposition of a system into its parallel dialogue and computational components.

Holistic design. There are so many aspects to the design of a large software system that it is usually extremely difficult to deal with those issues independently of one another. In particular, a holistic methodology supports design of both dialogue and computational components and serves both dialogue author and programmer roles. It integrates human factors into design lifecycles, provides automated support tools, and applies at all lifecycle phases including requirements/design and maintenance phases. Because of holistic design, communication among support tools is facilitated, and the system has a uniform author/programmer interface.

Meta-communication model. The initial approach of this work involved a combination of human factors and computer science specialists to produce optimal human-computer interfaces. We have found that this approach was not complete; communication is a major problem to be solved, and therefore, communication theorists must be included in research teams working on this problem. Communication theory says, for example, that messages exist on a great number of different levels, there being a primary problem-solving level (edit a file, land an airplane, book an airline reservation, etc.) and a communication management level to help make solving the problem easier. Some messages that human beings exchange refer directly to the meta-level communication management system itself, rather than directly to the primary problem at hand.

Cognitive structures. The emphasis of this project was a behaviorist approach to human factors issues in the human-computer interface. This approach is incomplete. Humans use computers to solve problems, and solving problems requires cognition. Although our behavioral research was successful in capturing expertise and using this information to aid the novice user, this empirical approach does not provide an understanding of the underlying cognitive structures. Cognitive modeling of computer users is required to provide this level of understanding.

User models for commanding computer action. This modeling effort was primarily methodological and was directed toward the use of clustering algorithms as a means of empirically capturing models of users commanding computer action in a text editing environment. Clustering algorithms were used to develop expert and naive user's models using the metaphor of a paper-and-pencil format instead of an actual interactive environment. The results of the model development showed high agreement among users suggesting that clustering algorithms may be a useful method of capturing the organizational structure of user command sets. In the cluster analyses, both the naive and expert users concurred on the four general edit tasks, and within each group users agreed on specific commands within each of those tasks.

The results of the validation study pointed out the possibility that certain aspects of interactive editing were not successfully modeled by the present approach. The large discrepancy between paper-and-pencil and interactive editing in the areas of mode change and current line location indicate that these are two deeper aspects which deserve more attention. Further research in interactive computing environments is needed to test the usefulness of these statistical procedures in representing the user's model in an interactive system.

A model for dynamic human-computer task allocation. The use of a modeling approach in a dynamic, multiple task, decision making situation was investigated. A model of human performance could be used to reduce a human's workload in a system in which the human and computer make decisions in parallel. The model would be used in place of explicit dialogue between human and computer as a method of avoiding redundant actions.

In order to model human event detection performance, it was necessary to determine the probability that the human would respond that an event had occurred. The event detection stage of the model used discriminant analysis to generate the probabilities that a human would classify a display into an "event detected" group or an "event not detected" group. These probabilities were based on observations of earlier performance by the same individual. The output from the first stage of the model was a vector of "event detected" probabilities, each entry in the vector representing a different system component. The second stage of the model assumed that the human acted as a stochastic controller. Dynamic programming was used to determine the optimal action at a specific point in time, and over a finite horizon. To determine this action, the probabilities of events were combined with the costs associated with alternative actions. The output of this stage of the model was the appropriate human action.

Two model validation experiments were conducted in a process control environment. The first experiment assigned no costs to actions. Thus, the user could simply act as an event detector. Discriminant analysis was descriptive of human event detection performance, predicting over 80% of the actions taken. The second experiment associated costs with actions. The model predicted nearly 85% of users' actions. The model predicted human performance over a range of parameter combinations, suggesting the possibility of its generalization to other situations.

Dialogue Design Tools

Three general categories of dialogue design tools evolved throughout the course of this project. These tools included system services, programming tools, and dialogue authoring tools. Tools developed to aid the programmer include a programming environment, a graphical programming language, and a behavioral demonstrator of software interfaces. Most of the effort in dialogue design tool building, however, centered around tools used by the dialogue author. These tools included a rapid prototyping tool, expert consultation tool, an expert profile generation tool, a general dialogue design environment, a dialogue author's toolkit, and various tools for voice I/O.

System services. The system services include all the tools developed to facilitate the conduct of the behavioral research and the execution of the DMS environment. These include the dialogue database, metering, internal communications, process creation and deletion, device input/output, timing, and debugging procedures.

Programming environment. An interactive Pascal programming environment (PEEP) was developed to demonstrate how the programmer's interface requires individualized human engineering. Using a contour model of program representation, PEEP provides an efficient human interface for program development that can be used to produce efficient object code. Traditionally production software is developed in an environment that is not human factored because compiled languages contain constraining features. For this reason, noncompiled languages, such as BASIC and LISP, are traditionally used in interactive environments. Although PEEP was only implemented in prototype form, its underlying contour model was developed to allow efficient code generation and interactive execution.

Graphical programming language. One the major failings of current methodologies is that there is no mapping between specification, simulation, and implementation phases. Through the use of a graphical programming language, specifications flow directly to implementations, and no textual coding takes place. As a system specification is decomposed to its finest details, worker functions appear that express the straight-line computational code for the computational components, and the dialogue components are completed using the dialogue author's interactive design environment (AIDE).

Behavioral demonstrator. This methodological tool permits system designers to test out or simulate system designs before the detailed system specifications are complete. Thus, in the flow from specifications to implementation there is no need to do a simulation or any of the associated recoding, and the result is a significant improvement in productivity and maintainability.

Rapid prototyping. Rapid prototyping and expert system tools for advising users and dialogue authors can increase the speed of construction and usability of the interface. Many companies have recently entered the rapid prototyping field with flashy dialogue author prototyping tools. Power, not flash, however, is what dialogue authors need. Specifically, the semantic power, rather than just concentration on a clever dialogue author interface, is needed. Representations of dialogue must be uniform (no supplementation by FORTRAN code), have full semantic power, be easy to edit, be executable, and be very high level (powerfully concise). A representation language must allow experimenters to implement theories of dialogue imposed on top of solving whatever problem the system is solving (editing, "painting" a graphical image, recording inventory, etc.). The rapid prototyping tool, IMMEDIATE, was constructed to achieve these goals.

Expert consultation tools. Human factors researchers have produced many interaction principles from carefully controlled experiments. We have found, however, that rigorous application of the principles is difficult, sometimes impossible. The principles are often stated in a rule-like form suggesting that embodying these principles in expert systems would lead to a more thorough and correct application. Initial explorations of a visual design consultant showed that building such an expert system would be quite difficult.

Expert profile generation. In order to define an automated file search assistant, a target profile methodology for representing expertise was developed. This methodology led to the development of a tool which can be used to generate a target profile which defines the selection of search procedures for a group of highly trained expert users and is based on a polling procedure. This polling procedure proved to be effective in differentiating file search strategies for novice and expert users of computer systems. Additionally, suggestive expert aiding based on this profile methodology was effective in improving novice users' search strategies.

Dialogue design environment. A software system called Generic ENvironment for Interactive Experiments (GENIE) was developed as a test-bed for human-computer interaction studies. GENIE contains many of the characteristics deemed necessary in such an environment, including graphical presentation of data, several simultaneous user tasks, interdependent interactions, and user required computations. In addition to these characteristics, GENIE was built to allow the human-computer dialogue to the system to be changed with relative ease.

Dialogue author's toolkit. The role of the computer scientist is to build tools for other professionals, and a major task that has been undertaken is the design of a specialized toolkit for the dialogue author. The goal of the toolkit is to allow the dialogue author to think functionally about the design of a human-computer interface without concern for the way in which the implementation is carried out. The toolkit must be powerful and capable of implementing the vast variety of possible interfaces. Moreover, the toolkit must be intelligent enough to promote the selection of good designs, and it must contain adequate tools for analyzing performance data collected while the interface is in use.

Standardized evaluation of speech recognition. The statistical rather than acoustical basis for current speech recognition hardware necessitates that distinctive vocabularies and optimal parameter settings be determined empirically. Central-composite design was demonstrated as a useful procedure to evaluate the performance of speech recognition systems under a variety of conditions and to predict recognition performance anywhere within the range of the variables studied. The use of this procedure enables the researcher to accomplish both a systematic evaluation of the effects of several variables simultaneously and the development of predictive tools that can aid designers in the implementation of speech recognition in human-computer interfaces. In some cases these empirical models may assist in the development of theoretical models of auditory information processing.

Other voice tools. In order to use the voice I/O hardware it was necessary to develop a variety of software tools including a phoneme editor for developing vocabularies for presentation using phoneme-based speech, two drivers for the voice recognition hardware, and statistical procedures to analyze and select vocabularies for voice recognition. Although these tools have not as yet been integrated into DMS, they are compatible with the DMS operating environment.

Dialogue Principles

The results of many of the behavioral studies not only were used to evaluate dialogue models and concepts, but they also provided a database to aid in the development of empirically derived dialogue design principles. Our research focused on three general areas of dialogue design principles which were particularly lacking in research data. First, new dialogue technologies using voice input/output were considered. Principles for voice output included message cueing and syntax and the use of visual redundancy. Design principles for voice input were concerned with user feedback and error correction. The second area of concentration was concerned with sophisticated dialogue design for novice end users. Screen layout principles for interactive window size as well as principles for providing novice user assistance in terms of interactive help and expert aiding were investigated. And, third, dialogue principles for dynamic human-computer task allocation were developed. These included both dialogue-based task allocation principles for command types and input device types as well as model-based task allocation principles dealing with model implementation and computer feedback.

Message format. The appropriate use of speech rather than visual displays in human-computer communication necessitates an understanding of the variety of information used by the listener to transfer information from the acoustic to the lexical level. These include syntactic processes, semantic processes, lexical processes, feature extraction, parametric processes, and environmental processes. Prior research with prerecorded emergency messages indicated that an alerting cue was needed prior to urgent spoken messages. However, phoneme-based synthesized speech has its own unique quality, a feature that may be processed by the listener and serve an alerting function. Our findings confirm that the unique quality of the synthesized speech serves the necessary alerting function if speech is used only for urgent messages. In this case no alerting cue should be used to minimize the response time to the emergency. Because high quality digitized speech does not have a unique speech quality, an alerting cue may be required even when speech is used only for warnings. In addition, when synthesized speech serves more than one function, the speech quality no longer serves to code urgent information. Then augmenting urgent information with an additional feature such as an alerting cue increases the detection rate of the critical messages. It should be noted that the use of an alerting cue prior to urgent messages in multiple-function speech systems yields a tradeoff between probability of message detection and time to respond to the emergency. In addition, no advantage is gained by using a keyword (e.g., emergency) in place of the alerting tone.

Another area in which the difference between auditory information processing and visual information processing affects the design of human-computer interfaces using speech displays is in the syntax for spoken messages. With visual displays, messages are usually presented in a terse format and the viewer has little difficulty determining the message. However, with auditory displays the listener uses linguistic information to reduce uncertainty about how to encode the message and determine meaning. Our research demonstrated that response latency to spoken emergency messages is shorter when messages are presented in a sentence format rather than in a keyword format with no syntactic redundancy. Continued research on auditory versus visual information display is critical to reveal these fundamental differences in information processing that affect the design of effective human-computer dialogues.

Redundant information. Speech can be used as a unique or a redundant source of information in a human-computer dialogue. The concept of cue summation from learning theory predicts that if cues from different modalities elicit the same response, the cues summate and increase the probability of the response. However, our research in the GENIE environment, which represents a high visual workload situation, indicated that no additional advantage over an auditory presentation was gained by using speech as a redundant source of information. Transcription accuracy was better with both the auditory and redundant presentations of selected types of information as compared to a completely visual task. However, performance in the primary task of directing aircraft to landing was better when the information was divided between the auditory and visual channels. With information divided across the two information channels, the auditory displays were alerting without distracting from the primary task, and the information presented visually allowed spatial comparison and referability. Presumably in the redundant presentation of messages users were distracted from the primary (visual) task while searching for the visual presentation of the messages. Research should be continued to determine how information should be allocated to the visual and auditory channels.

User feedback and error correction. In even the best case the empirical models developed using central-composite design predicted some speech recognition errors. These errors augment the usual human error in performing a task. Assuming that accuracy of data entry is important, the provisions for error prevention, error detection, and error correction are essential.

Recognition errors in voice entry dialogues have a multiplicative effect. The probability of a command begin correctly entered is a function of the recognizer accuracy and the length of the command. If p is the average probability that a recognizer will successfully classify an input utterance, then in a message of length N words the probability of the transmitted message being correct is p^N. Unless there is perfect recognition, the probability of a correct message will decrease rapidly. This effect was demonstrated in our research. Therefore, message length should be minimized to reduce command errors.

Another form of error prevention is to query the talker whenever the two top choices of the recognizer are sufficiently close that a decision cannot be made easily. This additional dialogue with the talker increased the percentage of correctly entered commands from approximately 65% to 75% with no significant increase in the time to enter the data.

User feedback and error correction is an approach to reducing error in systems using voice recognition. In our research the number of multiple-word commands or data fields correctly recognized was increased from approximately 65% to 95% with the addition of user feedback and error correction. Immediate feedback after each word spoken appears to be most useful. In fact, if feedback is given in the auditory channel and delayed until the complete field or command is entered, the time to enter the information will be doubled. In addition, error correction with auditory field-level feedback often takes the form of a complete command cancellation indicating the degree of confusion induced by this form of feedback. If feedback is given word-by-word, there is no advantage of visual over auditory feedback unless a large amount of visual display space is devoted to providing a history of the feedback. When feedback is provided word-by-word, users generally correct errors immediately using a single-word deletion. A complete command reset or cancellation is used only when the talker is totally confused. Users find a voice command editor difficult to use and invoke it only to provide additional feedback.

Finally, if the data to be entered using voice recognition are syntactically constrained, the use of automatic error correction should be considered. For example, with little more than the language parser used to process commands, approximately 50% of the errors transmitted by the recognizer were corrected. This reduction in errors would appear to the talker as an improvement in the accuracy of the speech recognition hardware. It should be noted that using the parser to correct errors results in some messages that are syntactically correct but not semantically correct. Whenever critical messages are being sent to the system, they should first be confirmed by the user to avoid disastrous system changes. Until speech recognition technology advances to a point where perfect recognition is attainable, investigation and application of error correction methods is an immediate and practical way of obtaining both adequate throughput rates and user acceptance in speech input tasks.

Window size. During our investigations of interactive file search displays, four dedicated window sizes (1-, 7-, 13-, and 19-lines) were investigated. The one-line window increased search time and total operations in file search. The number of different search operations was also greater for a one-line window. Therefore, one-line windows in retrieval systems and interactive editors should be avoided.

Interactive help. Various configurations of providing user versus computer initiated, selected, and displayed help were evaluated as a means of providing novice users of interactive systems with online assistance. The findings of this study suggest that user-initiated and user-selected, hard-copy help yields the best performance with novice users. In this particular help configuration, the users spent most of their time browsing the help information and looking at a variety of topics contained in the help file. When the help information was available through hard-copy manuals, the users could compare help presented in the manuals to the editing task presented on their terminal. A dialogue principle based on the results of this study would suggest that help information be constructed such that the user can browse and compare the various information files.

Expert aiding. A series of behavioral studies were conducted to determine the differences between novice and expert users performing interactive file searches. Large differences occurred between these two user groups both in terms of time to complete file searches and in terms of strategies of conducting searches. These results suggest that novice users who are required to perform complex file searches can benefit from expert advice. Subsequently, the target profile methodology was used to provide assistance to novice users. The file search assistant facilitated the development of expert search strategies by novice users and improved novice users' performance during assisted, but unadvised, time periods. This facilitating effect was particularly present during the use of more sophisticated file searching procedures. In terms of advice, however, the communication between the assistant and novice users was intrusive. To overcome this limitation, the implementation of a suggestive assistant with a strict diagnostic model is recommended for future development.

Dynamic human-computer task allocation. Tasks can be allocated dynamically by the human or the computer. Intuitively, it would be more motivating (but more work) for the human to assume the active role. In either case, human-computer communication is essential to inform the other party when and where one will allocate attention. Two types of communication can be utilized to convey the messages: explicit or dialogue-based communication and implicit or model-based communication.

Dialogue-based task allocation. Dialogue-based communication signifies the process in which the human uses some kind of computer input mechanism to relate intentions explicitly to the computer. To understand better and utilize dialogue-based communication for dynamic task allocation, a study was conducted to determine how human-computer system performance is affected by different input media and task allocation strategies. The task environment represented a simplified version of an air traffic control scenario wherein computer aid could be evoked by the human to accomplish task sharing between human and computer. Either dedicated function keys or the standard QWERTY keyboard was used as the medium to input commands. Four task allocation strategies were studied:

(1) Assignment by designation, wherein the computer might be requested to take over certain planes pinpointed by their associated identification numbers.

(2) Spatial assignment, wherein planes within a certain spatial confine were assigned to the computer.

(3) Temporal assignment, wherein planes within a certain time frame were assigned to the computer.

(4) Contingency-based assignment, wherein responsibilities for certain contingencies were assigned to the computer.

There was a significant effect of input medium on performance. Both performance and subjective preference measures indicated that function keys were the more desirable tool for input communication. In general, when function keys were used, more planes were landed and fewer errors were made. The effect of task allocation strategy on performance was also significant. Spatial assignment, contingency-based assignment, and assignment by designation achieved the highest levels of overall system performance. Temporal assignment was significantly poorer in this regard. Subjective ratings revealed an overall preference for assignment by designation, followed by spatial assignment and contingency-based assignment. Spatial assignment was the most powerful strategy in that this strategy could be used to allocate many tasks with one command. It was the least specific strategy, in that it could not be used to pinpoint particular tasks for assignment to the computer. Assignment by designation was the most flexible strategy in that it could be used to assign virtually any subset of the current tasks to the computer. It was also the most specific strategy. In terms of system performance and system adequacy data the two strategies were comparable.

Model-based task allocation. Model-based communication uses models of human performance to enable the computer to work cooperatively with the human in a reasonably conflict-free fashion. The computer uses these models to predict what the human is likely to do next. The computer then attends to tasks which will be neglected by the human. Communication is implied, and the computer utilizes these implicit messages to complement the human by averting conflicting or redundant actions. Model-based communication avoids the extra human workload associated with dialogue-based communication.

Following the development and experimental validation of a model of human decision making, an experiment was conducted in a process control environment in which a computer and human made decisions in parallel. In some conditions the computer would take the optimal action without regard to the human's actions (no model conditions). In other conditions the computer selected optimal actions based on the predicted actions of the human (model conditions). Crossed with model usage was the use of a unidirectional line of communication from the computer to the human. Use of the model improved human performance and system performance when compared to the conditions in which no model was used. This effect was most pronounced when unidirectional communication was not available. The model enabled the computer to work around the human, significantly reducing the number of redundant actions taken within the system. Unidirectional communication also significantly improved human and system performance. However, no cost was associated with unidirectional communication in the experiment. In a more realistic application, it is likely that communication of this type would temporarily divert the human's attention from the primary task and consequently degrade performance. In addition, when the computer communicates its actions to the human, the human is forced to work around the computer. Use of a model enables the computer to work around the human. Model-based communication was determined to be a reasonable means to achieve efficient dynamic task allocation in parallel human-computer systems. With the application of model-based communication, the need for dialogue-based communication becomes much less critical.

REPORTS AND ARCHIVAL PUBLICATIONS

Fifteen technical reports and more than thirty archival publications have been published as a result of this research program. Additional archival publications summarizing various aspects of the program are anticipated within six months of the termination date of the contract. Each of the published reports and papers is cited below with a brief description of its contents. The citations are divided into our four major technical sections: DMS - the software environment, the programmer, the dialogue author, and the end user.

1. DMS - The Software Environment

a. Reports

Ehrich, R. W. SAM -- A configurable experimental text editor for investigating human factors issues in text processing and understanding. Technical Report CSIE-81-3, September 1981 (AD A109331).

> This report describes the operation and internal structure of a text editor called SAM. SAM is constructed with modular parsers and functional components so that investigators can customize SAM to test particular issues in human-computer interaction.

Ehrich, R. W. A two-dimensional core graphics system for research in human-computer interfaces. Technical Report CSIE-81-4, October 1981 (AD A109280).

> This report is a users' guide to a partial implementation of a standard CORE graphics system designed to serve a number of inexpensive graphics devices in a consistent way. In order to facilitate studies in human-computer interaction in which performance is important, special functions have been added to enhance performance.

Roach, J. W., & Fowler, G. S. Rule-based programming for human-computer interface specification. Technical Report CSIE-82-5, January 1982 (AD A113036).

This report describes an implementation of a rule-based language related to PROLOG for the specification of human-computer interfaces. The specification of a human-computer communication requires a language in which the interface is expressed.

Ehrich, R. W. The DMS multiprocess execution environment. Technical Report CSIE-82-6, April 1982 (AD A117660).

This report is a complete users' guide to the DMS multiprocess execution environment. The report discusses the rationale for the use of a multiprocess environment and details many of the design issues.

Yunten, T., & Hartson, H. R. Human-computer system development methodology for the dialogue management system. Technical Report CSIE-82-7, May 1982 (AD A118287).

This report presents a preliminary version of the DMS holistic system development methodology. This methodology features a parallel work environment for the disciplined approach of the software engineer and the user-oriented approach of the human-factors engineer.

Lindquist, T., Fainter, R., Guy, S., Hakkinen, M., & Maynard, J. GENIE - A computer-based task for experiments in human-computer interaction. Technical Report CSIE-83-10, October 1983 (AD A137473).

A generalized task environment that contains elements appearing in several systems having human-computer interfaces is described. The software components are described at a functional level to provide the background for a discussion of how the human-computer interface for GENIE can be created. An example of using GENIE for a voice output experiment is provided.

b. Archival Publications

Roach, J. W., Hartson, H. R., Ehrich, R. W., Yunten, T., & Johnson, D. H. DMS: A comprehensive system for managing human-computer dialogue. In *Proceedings of Human Factors in Computer Systems*. (pp. 102-105). Washington, DC: Association for Computing Machinery, 1982.

> This paper describes research being carried out to construct DMS (Dialogue Management System), a complete system for defining, modifying, executing, and metering human-computer dialogues. DMS has three major aspects--the technical environment for software production and execution, a set of software design tools, and a methodology for software design. This paper surveys all three aspects.

Ehrich, R. W. DMS - A system for defining and managing human-computer dialogues. In G. Johansen, & J.E. Rijnsdorp (Eds.), *Proceedings of IFAC/IFIP/IFORS/IEA Conference on Analysis, Design, and Evaluation of Man-Machine Systems*. (pp. 367-375). Geneva: International Federation of Automatic Controls, 1982. (Also *Automatica*, in press.)

> DMS is a complete system for defining, modifying, simulating, executing, and metering human-computer dialogues. This paper discusses the concept of dialogue independence, which is one of the reasons for selecting a multiprocess execution environment for DMS. Issues of software decomposition are discussed, and the multiprocessing constructs are presented.

Ehrich, R. W. DMS - An environment for building and testing human-computer interfaces. In *Proceedings of the International Conference on Cybernetics and Society*. (pp. 50-54). New York: IEEE Systems, Man and Cybernetics Society, 1982.

> This paper discusses the DMS execution environment and concentrates on the input/output services that are supported by the environment. These services are the building blocks upon which the dialogue author's toolkit are based, and they provide for sophisticated levels of dialogue concurrency.

Roach, J. W. A methodology for structuring the logic of interactions in human terms. In *Proceedings of the Human Factors Society 27th Annual Meeting*. (pp. 860-864). Santa Monica, CA: The Human Factors Society, 1983.

> Creating systems that contain "simple" and "clear" logical sequences is an art, perhaps even an accident, rather than a well-understood science. In this paper artificial intelligence techniques are applied to create systems with a very fluid, dynamic problem-solving structure to prevent rigid, inflexible logical sequencing.

Roach, J. W., & Nickson M. Modeling and developing human/computer communications. In *CHI '83 Conference Proceedings: Human Factors in Computing Systems* (pp. 35-39). New York: The Association for Computing Machinery, 1983.

This paper describes a new method for modeling, designing, and developing dialogues, a method that has a strong formal basis and allows a uniform syntactic and semantic specification. This formal descriptive technique has the added advantage of being executable, that is, it has widely available translators.

Yunten, T.; & Hartson, H. R. A SUPERvisory methodology and notation for development of human-computer systems. In *Advances in human-computer interaction*. Norwood, New Jersey: Ablex Publishing Corp., in press.

The SUPERvisory Methodology And Notation (SUPERMAN) is a new human-computer system development methodology to enable system developers to evaluate a system's human-factors, and supports the roles of dialogue author and programmer. Its single representation technique produces an executable software requirements specification. The methodology is built into a system development environment supported by an integrated set of automated tools.

2. *The Programmer*

a. *Reports*

Ku, C., & Lindquist, T. E. PEEP: A Pascal environment for experiments on programming. Technical Report CSIE-82-9, September 1982 (AD A137474).

This paper describes a Pascal interactive programming environment called PEEP which has an internal program representation that is based on a semantic model of computation. The model includes a flexible data structure that acts as a common object for integrating different software tools into the environment. PEEP includes features such as source level debugging, alternative identifier binding strategies, and program statement level interaction with the user.

Kafura, D., Lee, J. A. N., Lindquist, T., & Probert, T. Validation in ADA programming support environments. Technical Report CSIE-82-12, December 1982 (AD A124765).

This report addresses the processes for Ada Programming Support Environment (APSE) implementation in terms of the Ada Programming Language and uses those specifications to suggest a mechanism for validation development. It also discusses how an extended model would provide a fundamental basis for the design of Ada systems which respond to the need to provide networking, distributed processing, and security enclaves.

b. Archival Publications

Lindquist, T. The application of software metrics to the human-computer interface. In *Proceedings of the IEEE COMPCON FALL '83 Conference* (pp. 239-244). New York: IEEE, 1983.

This paper presents a technique that can be used to evaluate alternative human-computer dialogue designs at the control structure level. The evaluation focuses on the effort-to-learn and effort-to-use the structure underlying a human-computer interface. An extended example is given in which the programmer's dialogue to a syntax-specific editor is compared to that of a more typical interactive system. Well known software metrics are used in this analysis.

3. The Dialogue Author

a. Reports

Johnson, D. H., & Hartson, H. R. The role and tools of a dialogue author in creating human-computer interfaces. Technical Report CSIE-82-8, May 1982 (AD A118146).

A new human-computer system design role is introduced. A dialogue author is responsible for creating the dialogue which constitutes the human-computer interface of an application system. The role of the dialogue author and the tools of the Author's Interactive Dialogue Environment (AIDE) are discussed.

Hoernemann, J. W., Maynard, J. F., & Williges, B. H. Software tools for voice recognition research, Technical Report CSIE-82-11, November 1982 (AD A124447).

This report describes software tools provided to interface a voice recognition device to a VAX 11/780 computer. The purpose of the tools is to provide a flexible and easy-to-use set of software modules for experimenters to implement built-in hardware functions of the recognizer and to analyze vocabularies using statistical procedures.

Hartson, H. R., & Johnson, D. H. Dialogue management: New concepts in human-computer interface development. Technical Report CSIE-83-13, November 1983 (AD A136945).

Dialogue management is an emerging field which emphasizes development of quality human-computer interfaces. Key concepts in dialogue management are surveyed, and their importance to overall system design is discussed. These concepts include dialogue independence, the role of a dialogue author, and a holistic methodological approach to system development.

Callan, J. The behavioral demonstrator: A requirements executor. Technical Report CSIE-83-14, May 1983 (AD A136944).

This report presents a design for tools which aid in the requirements verification of computer systems. These tools use a very high level graphical requirements specification language and a system development methodology for human computer systems. The report moves from an abstract design to actual implementation and uses a sample application system throughout for illustration.

Johnson, D. H., & Hartson, H. R. Issues in interactive language specification and representation. Technical Report CSIE-83-15, November 1983 (AD A137478).

Issues in the implementation of interaction languages are discussed, including language design, specification, representation, and recognition. Components of an interaction language are taxonomized and a model for interaction language specification is presented. An interactive example-based interface for interaction language specification called Language-By-Example (LBE) is introduced along with a sample scenario of its use.

b. *Archival Publications*

Roach, J. W., Pittman, J. A., Reilly, S., & Savarese, J. A visual design consultant. In *Proceedings of the International Conference on Cybernetics and Society* (pp. 189-193). New York: IEEE Systems, Man and Cybernetics Society, 1982.

In this paper a visual design tool to aid programmers in developing better human-machine dialogues is described. We hypothesize that incorporating into dialogues design principles that are based on behavioral studies of human information processing capabilities will improve the quality of human-machine interaction.

Cohill, A. M., & Ehrich, R. W. Automated tools for the study of human/computer interaction. In *Proceedings of the Human Factors Society 27th Annual Meeting* (pp. 897-900). Santa Monica, CA: The Human Factors Society, 1983.

In order to facilitate the analysis of behavioral data and to obtain the maximum amount of information from experimental data, automated tools are required in order to assist the investigator in the data analysis task. This paper discusses the need for standardized data formats and gives an example of a simple yet flexible tool that is now being routinely used.

Spine, T. M., Williges, B. H., & Maynard, J. F. An economical approach to modeling speech recognition accuracy. *International Journal of Man-Machine Studies*, in press.

A central-composite design methodology is recommended as an economical means to develop empirical prediction equations for voice recognizer performance incorporating a number of influential factors. The paper demonstrates how the empirical models developed can be incorporated into a design tool for the dialogue author to optimize the percentage of correction recognitions. In addition, the methodology is recommended as part of a standardized test procedure for voice recognition equipment.

4. *The End User*

a. *Reports*

Greenstein, J. S., & Revesman, M. E. A Monte-Carlo simulation investigating means of human-computer communication for dynamic task allocation. Technical Report CSIE-81-1, August 1981 (AD A103890).

This paper proposes that tasks in certain human-computer systems be allocated in a dynamic manner. Simulation experiments investigate two means of communication for dynamic task allocation: implicit, in which the computer uses a model of human decision making to infer the human's actions, and explicit, in which the human overtly communicates his action plans to the computer.

Williges, B. H., & Williges, R. C. User considerations in computer-based information systems. Technical Report CSIE-81-2, September 1981 (Revised January 1982) (AD A106194).

This report compiles over 500 dialogue author considerations for the design of human-computer interfaces. Topics covered include data organization, dialogue modes, user input devices, feedback and error management, security and disaster prevention, and multiple user communication.

b. *Archival Publications*

Greenstein, J. S., & Revesman, M. E. A Monte-Carlo simulation investigating means of human-computer communication for dynamic task allocation. In *Proceedings of the International Conference on Cybernetics and Society* (pp. 488-494). New York: IEEE Systems, Man and Cybernetics Society, 1981.

See Technical Report CSIE-81-1 for details.

Greenstein, J. S., Williges, R. C., & Williges, B. H. Human-computer dialogue design: Hardware and software. In *Proceedings of the 1981 Fall Industrial Engineering Conference* (pp. 89-98). Atlanta, GA: Institute of Industrial Engineers, 1981.

This paper outlines the research and design implications of the human's role as a supervisory controller within increasingly automated systems. Successful implementation of these systems involves appropriate design and management of human-computer dialogue. The paper focuses upon the hardware and software components that combine to specify human-computer dialogue.

Folley, L. J., & Williges, R. C. User models of text editing command languages. In *Proceedings of Human Factors in Computer Systems* (pp. 326-331). Washington DC: Association for Computing Machinery, 1982.

User models of command language selection were developed for both naive and expert users of text editors. These models were empirically derived from paper-pencil edits using statistical clustering algorithms. Naive and expert users showed agreement in terms of using four edit class clusters but demonstrated marked differences within their command clusters.

Williges, B. H., & Williges, R. C. Structuring human/computer dialogue using speech technology. In *Proceedings of Workshop on Standardization for Speech I/O Technology* (pp. 143-151). Gaithersburg, MD: National Bureau of Standards, 1982.

Three approaches to generating data for the development of dialogue design guidelines for the use of speech devices are reviewed. These include using data from research with other auditory equipment, building theoretical and empirical models with current speech hardware, and predicting system requirements using simulated hardware. Areas requiring human factors research are noted.

Revesman, M. E., & Greenstein, J. S. Stochastic control applied to human-computer interaction. Preprint distributed at the TIMS/ORSA Joint National Meeting, Detroit, Michigan, April 1982.

This paper presents an approach to modeling the human acting as a stochastic controller in a dynamic, multiple task situation. Such a model could be used by a computer acting as a decision maker in parallel with the human. The computer could determine an optimal strategy given the model's predictions of future human actions.

Williges, R. C. Design of human-computer dialogues. In *Proceedings of the 8th Congress of the International Ergonomics Association* (pp. 78-79). Santa, Monica, CA: International Ergonomics Association, 1982.

An integrated conceptual approach to the design of human-computer dialogues for novice computer users is presented. The approach involves the combined efforts of human factors researchers and computer scientists. Various user considerations are reviewed, and several human factors research issues directed toward obtaining empirically based design principles are discussed.

Williges, R. C., & Williges, B. H. Human-computer dialogue design considerations. In G. Johansen, & J. E. Rijnsdorp (Eds.) *Proceedings of IFAC/IFIP/IFORS/IEA Conference on Analysis, Design, and Evaluation of Man-Machine Systems* (pp. 273-280). Geneva: International Federation of Automatic Controls, 1982. (Also *Automatica*, 1983, *19*, 767-773.)

> This paper discusses the requirement for empirically based dialogue design guidelines. Results of two studies directed toward specifying dialogue principles are presented. One study deals with a methodology for developing user models, and the second study deals with the design of HELP information. Computer-aided implementation of dialogue guidelines is discussed as an implementation tool.

Elkerton, J., Williges, R. C., Pittman, J. A., & Roach, J. W. Strategies of interactive file searching. In *Proceedings of the Human Factors Society 26th Annual Meeting*, (pp. 83-86). Santa Monica, CA: The Human Factors Society, 1982. (Also *Human Factors*, in press.)

> Novice and expert computer users were compared in terms of their interactive file search strategies and use of interactive display window sizes. Search procedures evaluated included: scrolling, paging, string search, absolute line movement, and relative line movement. Marked differences occurred between novice and expert users in terms of search times and strategies thereby suggesting the need for on-line aiding.

Hakkinen, M. T., & Williges, B. H. Synthesized voice warning messages: Effects of alerting cues and message environment [Abstract]. In *Proceedings of the Human Factors Society 26th Annual Meeting* (p. 204) Santa Monica, CA: The Human Factors Society, 1982. (Also *Human Factors*, in press.)

> This paper summarizes a study that examined the need for an alerting cue preceding synthesized speech warning messages where speech is used for either a single function or multiple functions. Guidelines that are based upon demonstrated performance differences with end users are provided for the dialogue author.

Folley, L. J. & Williges, R. C. Validation of user models for interactive editing. In *Proceedings of the Human Factors Society 26th Annual Meeting* (pp. 616-620). Santa Monica, CA: The Human Factors Society, 1982.

Previously developed user models based on clustering procedures were validated in an interactive editing environment. Differences in command frequency counts between the paper-pencil and interactive editing environments suggest that deeper aspects of interactive editing, such as mode change and current line location, must be included in user models.

Cohill, A. M., & Williges, R. C. Computer-augmented retrieval of HELP information for novice users. In *Proceedings of the Human Factors Society 26th Annual Meeting* (pp. 79-82). Santa Monica, CA: The Human Factors Society, 1982. (Also *Human Factors*, in press.)

Retrieval of HELP information in terms of initiation, selection, and presentation for novice computer users was evaluated. Overall, HELP information improved task performance. The results were discussed in terms of design principles to facilitate browsing and comparison features of on-line assistance.

Rieger, C. A., & Greenstein, J. S. The allocation of tasks between the human and computer in automated systems. In *Proceedings of the International Conference on Cybernetics and Society* (pp. 204-208). New York: IEEE Systems, Man and Cybernetics Society, 1982.

This paper integrates a dynamic task allocation approach with traditional methods of human-machine function allocation to develop a general procedure for human-computer task allocation. A flow diagram sequences the task allocation design decisions required when functions may be performed by both human and automated components of a system.

Revesman, M. E., & Greenstein, J. S. Human/computer interactions using a model of human decision making. In *Proceedings of the International Conference on Cybernetics and Society* (pp. 439-443). New York: IEEE Systems, Man and Cybernetics Society, 1982.

When human and computer perform similar tasks in parallel, an effective line of communication must exist between the two entities. An implicit method of communication is suggested in which the computer uses a model of human decision making to infer the human's actions. This paper develops a two-stage model of human performance that can be used to achieve implicit communication.

Sanford, D. L., & Roach, J. W. Evaluating natural language communication to improve human-computer interaction. In *Proceedings of the International Conference on Cybernetics and Society* (pp. 194-198). New York: IEEE Systems, Man and Cybernetics Society, 1982.

This paper reports the results of a study that investigates differences when novices use standard command language versus natural language with a computer interface. The results indicate that there are initially significant differences between natural language and command language conditions, but these differences quickly disappear as the people adapt to using the unfamiliar command language format.

Elkerton, J., & Williges, R. C. Development of an adaptive assistant in a file search environment. In *Proceedings of the Artificial Intelligence Conference*. Rochester, MI: Oakland University, 1983.

This paper reviews concepts of adaptive systems and describes the development of a methodology for empirically capturing expert file searching strategies. The method provides a composite profile of expert search strategies which can be readily implemented to provide on-line assistance.

Rieger, C. A., & Greenstein, J. S. The effects of dialogue-based task allocation on system performance in a computer-aided air traffic control task. *Behavior Research Methods and Instrumentation*, 1983, *15*, 208-212.

This paper reports a study of five dialogue strategies for dynamically allocating tasks between human and computer. Participants in the study acted as controllers in a simplified representation of an air traffic control system, sharing their tasks with a computer controller. The results provide implications for the design of effective task allocation dialogues.

Spine, T. M., Maynard, J. F., & Williges, B. H. Error correction strategies for voice recognition. In *Proceedings of the Voice Data Entry Systems Applications Conference*. Palo Alto, CA: American Voice Input/Output Society, 1983.

The use of current voice recognition hardware introduces error into the data entry dialogue. This paper describes a study where several different end-user approaches to error correction were examined in both event-based and time-based tasks. The effectiveness of user-initiated correction is compared to computer-initiated error correction.

Revesman, M. E., & Greenstein, J. S. An empirical validation of a model of human decision making for human-computer communication. In *Proceedings of the Human Factors Society 27th Annual Meeting* (pp. 958-962). Santa Monica, CA: The Human Factors Society, 1983.

This paper validates a two-stage model of human decision making in a multiple task process control situation. The model permits implicit communication between human and computer, wherein the model is used by the computer to infer the human's probable actions. The model proves to be an accurate predictor of human performance in the situation studied.

Schurick, J. M., Maynard, J. F., & Williges, B. H. Feedback techniques for voice input in computer-based systems [Abstract]. In *Proceedings of the Human Factors Society 27th Annual Meeting* (p. 102). Santa Monica, CA: The Human Factors Society, 1983.

This paper reports a study to examine user feedback for error detection when voice recognition is used for data entry. Both auditory and visual forms of feedback were examined. A dialogue to confirm entries whenever a clear selection cannot be made by the recognizer was also investigated. The results are discussed in terms of source data accuracy and time required by the end user to enter and correct the information.

Elkerton, J., & Williges, R. C. Evaluation of expertise in a file search environment. In *Proceedings of the Human Factors Society 27th Annual Meeting* (pp. 521-525). Santa Monica, CA: The Human Factors Society, 1983.

A target profile methodology was used to evaluate the file searching strategies of novice and expert computer users. Expert and novice users differed both in terms of search times and search strategies. The profile methodology was discussed as a useful tool in capturing expert search strategies.

Revesman, M. E., & Greenstein, J. S. Application of a model of human decision making for human/computer communication. In *CHI '83 Conference Proceedings: Human Factors in Computing Systems* (pp. 107-111). New York: The Association for Computing Machinery, 1983.

This paper employs a two-stage model of human decision making in an experimental situation in which both a human and a computer act as decision makers. The model is used to achieve implicit communication between human and computer, wherein the computer infers the human's probable actions. Implementation of the model significantly improves the human's performance and overall system performance without degrading the computer's performance.

Reilly, S., & Roach, J. W. Improved visual design for graphics display. *IEEE Computer Graphics and Applications*, in press.

The principles of visual design employed in the field of advertising are applied to improve visual displays for human-computer interfaces. The principles are first explained and then examples of displays using an air traffic control display and an airline reservation system are shown.

Williges, B. H., & Williges, R. C. User considerations in computer-based information systems. In F. A. Muckler (Ed.) *Human Factors Review 1983*. Santa Monica, CA: The Human Factors Society, in press.

See Technical Report CSIE-81-2 for details.

TECHNICAL INFORMATION EXCHANGE

In order to provide an adequate information exchange so that our research would be timely and relevant to operational and proposed Navy systems, numerous briefings occurred between our project team and Navy scientific advisors as well as other members of the scientific community involved in human-computer dialogue research. Members of the project team participated in over 75 formal and informal briefings.

Effective coordination mechanisms were established with various Navy laboratories that included members of our scientific advisory board. Formal briefings were provided to the advisory board every six months during the project. In addition, to accomplish our goal to provide relevant research the following Navy laboratories were contacted one or more times.

Naval Air Development Center (NADC)
Naval Ocean Systems Center (NOSC)
Naval Personnel R&D Center (NPRDC)
Naval Postgraduate School
Naval Research Laboratory (NRL)
Naval Surface Weapons Center (NSWC)
Naval Training Equipment Center (NAVTRAEQUIPCEN)
Naval Underwater Systems Center (NUSC)
Naval Sea Systems Command (NAVSEA)
Office of Naval Research (ONR)
Office of Naval Research, London Office (ONR-London)
Pacific Missile Test Center (PMTC)

Other governmental laboratories and departments contacted include:

Army Foreign Science and Technology Center
Army Research Institute (ARI)
Department of the Interior, Geological Survey
NASA-Ames Research Center
National Science Foundation
Wright-Patterson AFB, STARS Program

Another goal of our briefings was to provide an exchange of ideas between our project team and other scientists actively involved in human-computer interface design. These meetings included the following organizations and universities located in the United States:

Bell Laboratories
Carnegie Mellon University
Digital Equipment Corporation
George Washington University
Honeywell
Hunter College
International Business Machines
KAPSE Interface Team
Massachusetts Institute Technology
Mitre Corporation
National Academy of Engineering
National Academy of Sciences
North Texas State University
Ohio State University
Pennsylvania State University
Psycholinguistic Associates
Purdue University
University of Maryland
Xerox Corporation

Personnel from the following international organizations and universities were involved in briefings on our technical activities:

Forschungsinstitut fuer Anthropotechnik (FAT)
Hitachi Central Research Laboratories
Institute fuer Informationsverarbeitung in Technic and Biologie (IITB)
Loughborough University of Technology
Swedish Telecom Headquarters
University of Aston
University of Nottingham

One member of the project team, Joel S. Greenstein, participated in the Navy/ASEE summer faculty research program. During the summer of 1983 he joined the Human Engineering Branch (Code 8231) of the Advanced Command Center Technology Division, NOSC, where he participated in the development of a report on a lightweight modular display system and provided information for a proposal defining human factors research and development critical to successful implementation of the system. In addition, he designed and conducted research to evaluate the effects of three monochrome and four color automated status board display formats on operator task performance.

INTERDISCIPLINARY GRADUATE TRAINING

Project Workshops and Seminars

Periodically throughout the contract the project team conducted weekly seminars to discuss current project activities in detail, to develop plans for future research, and to discuss various human factors and computer science topics central to the research. In addition to all members of the project team, interested students from CS and IEOR attended these workshops. Location of the meetings alternated between CS and IEOR. In addition, several one-day intensive workshops were conducted to develop the generic task environment (GENIE), to exercise the DMS methodology, and to determine the necessary software tools and dialogue principles required by the dialogue author.

Graduate Courses

Several new graduate courses dealing with the human-computer interface were offered. These include CS 6690 (Advanced Topics in Programming Systems), CS 5332 (Information Storage and Retrieval II), and IEOR 6380 (Human-Computer Dialogue Design). These courses emphasized software metrics, program testing and proving, programming environments, dialogue description techniques, and human-computer interface dialogue principles. The senior-level human factors survey course (IEOR 4200) was modified to cover the human-computer interface in more detail. Finally, as a result of this joint research program increasing numbers of graduate students in both departments have elected courses in the other department as part of their graduate program of studies.

Graduate Research

Many of the specific research problems addressed in this research effort resulted in graduate theses and dissertations for students in both CS and IEOR. A total of thirteen graduate degrees completed or in progress are based on this SRO project. The faculty committees of these students were interdisciplinary comprised of faculty members from both departments. In addition, many graduate students not directly involved in this project have sought out committee members from other disciplines involved in the human-computer interface problem as a result of our joint research effort. Theses and dissertations completed or in progress are listed below.

Theses

Cohill, A. M. Information Presentation in Software HELP Systems, Master of Information Systems, Computer Science, VPI&SU, March 1982.

Guy, S. R. Design and Implementation of the Generic Interactive Environment's Experiment Profile Process and User Aircraft Control Process, Master of Information Systems, Computer Science, VPI&SU, March 1982.

Folley, L. J. The Development and Validation of a User's Model for Interactive Text Editing, Master of Science, Industrial Engineering and Operations Research, VPI&SU, May 1982.

Hoernemann, J. Design, Implementation, and Application of a Human/Computer Voice Input System, Master of Information Systems, Computer Science, VPI&SU, May 1982.

Ku, C. S. The Design and Implementation of a Language Environment for Evaluating the Programming Task, Master of Science, Computer Science, VPI&SU, May 1982.

Ahmed, S. A Multiprocess Execution Environment for a Dialogue Management System, Master of Science, Computer Science, VPI&SU, August 1982.

Narang, P. Dynamic Languages for Human-Computer Interaction, Master of Science, Computer Science, VPI&SU, November 1983.

Elkerton, J. An Experimental Evaluation of an Assistance System in an Information Retrieval Environment, Master of Science, Industrial Engineering and Operations Research, VPI&SU, January 1984.

Hakkinen, M. The Effects of Message Mode, Length, and Structure in a Multi-function Voice Synthesis System, Master of Science, Industrial Engineering and Operations Research, VPI&SU, in progress.

Dissertations

Revesman, M. E. Validation and Application of a Model of Human Decision Making for Human/Computer Communication, Ph.D., Industrial Engineering and Operations Research, VPI&SU, October 1983.

Johnson, D. A Human-Oriented Specification of Interaction Languages for Human-Computer Interfaces, Ph.D., Computer Science, VPI&SU, in progress.

Pittman, J. Predicting the Usefulness of Expert Consultant Systems with Information Theoretic Measures of the Level of Expertise, Ph.D., Industrial Engineering and Operations Research, VPI&SU, in prograss.

Yunten, T. A SUPERvisory Methodology and Notation for Development of Human-Computer Systems, Ph.D., Computer Science, VPI&SU, in progress.

PROGRAM IMPACT

Research Impact

A broad approach to human-computer dialogue design was followed throughout this research program rather than adopting a more focused investigation on one aspect of the dialogue design problem. Obviously such an approach results in less depth in investigating any of the four major areas depicted in Figure 1. However, the broader approach taken in this research program had a tremendous impact on defining and organizing the problems associated with human-computer communication interfaces.

The cross-fertilization of ideas from one aspect of the problem to the other major areas could only be realized through this broad approach. Indeed, there are probably no comparable laboratories with such broad perspective of the problem of producing software systems with human-computer interfaces. For example, the concept of role, training, and activities of the dialogue author could only have come from close interaction of computer and behavioral scientists. The tools and advanced techniques used in implementing studies could only have been achieved through the efforts of the computer science team members, and the recognition and resolution of the human factors issues in DMS could only have been achieved through daily work with the human factors team members. These relationships were manifest not only in the many joint seminars and workshops but also in daily contacts and activities.

The programmatic support of the SRO program provided us with the resources to consider such a large-scale approach. We were also able to capitalize on the individual areas of expertise of all the researchers in a synergistic fashion which, in turn, resulted in multidisciplinary solutions to the various research problems. Rather than forcing researchers who have recognized expertise in certain research areas to become expert in every associated field, our multidisciplinary approach resulted in sophisticated, interdisciplinary strategies.

Many of the graduate students working with the faculty on this research program benefited greatly from this synergistic approach. Not only did they gain an appreciation for the many facets of the human-computer software interface problem, they also realized the multidisciplinary views required to solve such problems. In addition, most of the graduate students developed a strong set of interdisciplinary research skills in both the computer and behavioral sciences. Their participation in this program encouraged them to take graduate courses and choose faculty advisory committees from both computer science and human factors discipline areas.

New Research Opportunities

One of the most encouraging aspects of the research program is the potential for extensions into new research and application areas. Several such opportunities with industry, the Navy, and the National Science Foundation were a direct outgrowth of this SRO program.

DEC internship program. Our industrial advisor on this effort was the Digital Equipment Corporation (DEC). Based on their knowledge of this research effort and the type of interdisciplinary graduate training that developed, the DEC Software Human Engineering staff initiated a new three-year Human Factors Graduate Intern Fellowship Program with Virginia Tech with Robert C. Williges serving as the program manager. This program will support graduate training in human-computer software interfaces and will allow graduate students to work both in the laboratories at Virginia Tech and in the laboratories at DEC in New England. Projects of common interest that will be pursued in this program include innovative approaches to computer-based information management and the development of iterative design methodologies for human-computer interfaces.

Navy transitions. Visits with our various Navy scientific advisors have given us the opportunity to share our research results with a variety of Navy laboratories. In particular, NSWC personnel have been extremely supportive of this research effort. This has had a direct impact in terms of courses being taught by our faculty at that facility and a continuing interest in the dialogue authoring tools and voice input/output research. Additionally, the visibility of this research program was instrumental in encouraging one Navy officer to pursue a Ph.D. research program in the human-computer interface area at Virginia Tech. His research centers on the problem of designing symbiotic human-computer systems, and his graduate program is interdisciplinary involving coursework in both human factors and computer science. His dissertation research is being supported by a contract from NADC.

NSF research program. A direct outgrowth of this research program is a new three-year National Science Foundation (NSF) research grant to the Computer Science Department at Virginia Tech with H. Rex Hartson and Robert W. Ehrich serving as co-principal investigators. This research effort will be directed toward still deeper understanding of the structure of human-computer interfaces and the associated linguistic issues.

Implications of Research Program

Although a great deal of research was completed during the three years of this SRO program, the results of our work suggest some additional areas of concentration. Several of these issues are described as examples of direct outgrowth of this research program for future research.

Dialogue management system. The main concept upon which the DMS is based is that of dialogue independence. It is manifest in the separation of dialogue-author and applications-programmer roles and tools; this is the key to quality human-engineered interfaces. Until now attention has necessarily been focused on interface design issues and DMS functionality. Implementation of DMS has progressed to the point where we are beginning to apply it to real design tasks. Two questions about our work need to be answered: is DMS capable of addressing the broad class of design issues we claim it will, and can it be used by non-computer professionals? These are nonparametric questions, and therefore cannot be tested using traditional controlled human-factors experimentation. DMS is so large in scope that it would take years to provide alternative hypotheses against which to test it. However, the above two questions are human performance issues, and methods of iterative refinement are feasible for the extraction of that information from dialogue authors and programmers who use the system.

Author's interactive dialogue environment. Central to the concept of a DMS is a methodology for designing large software systems in which human-computer interface design is identified as a separate activity. One consequence is that it is now possible to define a special role, that of the dialogue author, and to provide a kit of specialized tools for designing, testing, and modifying human-computer interfaces. Our experience so far indicates that in such an environment the complexity of the task of building quality interfaces is reduced.

The task of building an effective toolkit is challenging. For one thing, prototypes have to be developed through carefully planned behavioral-research studies. More difficult is the fundamental problem of modeling the dialogue processes the tools are required to support.

Future work ought to be concerned with the expansion of the capabilities of the tools in the authoring environment and the addition of new tools. A transaction model has been defined in this research program to account for the structure of basic dialogue sequences, but that model should be verified and expanded. Central to the transaction model is a language interpreter called DYLEX, and its functional capabilities should be evaluated and expanded to account for additional needs. Some potential needs include:

(1) Inclusion of more semantics in the transaction model;

(2) Provision for more general interaction types;

(3) Serial and parallel interactions;

(4) Dynamic interactions;

(5) Transaction and interaction networks;

(6) Multiple expertise levels; and

(7) Suspended, interruptable, and expiring transactions.

Multichannel concurrency. Another important aspect of a DMS that needs to be studied is multichannel concurrency. An interaction with concurrent dialogue over multiple channels (e.g., in which the user has an option of responding with some combination of typing, touching, and speaking) implies a complexity that few programmers would care to face. Within the framework of the transaction model separate design-time tools build separate definitions for the various channels; and, therefore, the execution environment and the transaction executor need to be designed in order to provide for the concurrent execution of those definitions.

Supervisory methodology. One of the cornerstones of DMS is the methodology by means of which system specifications were transformed, through DMS, to a final implementation. That methodology used a single, simple, design representation that incorporated both data flow and control flow, and produced requirement specifications directly verifiable by the customer/client/users. The methodology needs to be extended in several ways, and tools for its support in DMS need to be designed and implemented. Two primary areas of future work on methodology are: the development of a high-level graphical programming language and its translator and the construction of a behavioral demonstrator. The graphical programming language would provide rapid coding directly from the system specifications. The behavioral demonstrator, on the other hand, would allow the execution of the automated control structure to be observed by application programmers and studied by human-factors experts to verify the fulfillment of their system requirements.

Adaptive human-computer interfaces. Our research program emphasized the specification of human-computer dialogue-design principles optimized a given interface. An alternative is to optimize or satisfy with an adaptive interface in which the dialogue is tailored to the characteristics of the user, the level of experience of the user, and the task configuration. Research questions should be addressed on the type of assistance, monitoring of the assistant, frequency of use of the assistance, and form of dialogue with the assistant.

Bandwidth effects. A prevalent design assumption is that in order for human-computer interfaces to be more effective, they should mimic the human's natural world. If a human opens a book, the entire page is seen instantly; however with a machine of low bandwidth, it would take a great deal of time for the page to fill. Formal experiments are needed in applications such as database searching to determine whether in fact bandwidth is an important concept for human performance.

Multi-mode interfaces. Advances in touch-sensitive input devices and speech products for input and output make the use of those sensory modalities a more viable alternative to the keyed input and visual output traditional in human-computer dialogues. Although multi-media human-computer dialogues make exciting demonstrations, little research has been directed toward determining when such dialogues are appropriate and how to optimize their interfaces. To provide the maximum flexibility in the design of multi-media human-computer dialogues, more research is required to determine the capabilities and limitations of new input and output devices. For example, when voice recognition equipment is connected in parallel with the host processor, it is possible to exceed the vocabulary limitations of the hardware by rapidly uploading and downloading vocabularies from the host. Therefore, systematic investigations of human factors requirements for large vocabularies with voice-recognition equipment should be undertaken.

Verification and error processing. An important but often neglected aspect of human-computer interfaces is the verification of the user's actions and the subsequent error-processing and recovery that takes place upon negative verification. Certain types of verification can be defined directly in the syntax of the dialogue transaction, e.g., whether or not a user is typing a number with valid format and permissible range. During our research program, enhancement speech recognition was demonstrated with simple syntactical checks. However, general verification mechanisms are needed for the more complex cases such as those in which a database query is required after the user types a name. Whenever verification is negative, the problem becomes more complex. For example, should error messages and recovery procedures be specified as part of the syntax of the dialogue transaction, or should other dialogue transactions be initiated as a consequence of the faulty user-action?

Training novice users. Our research program focused on the design of human-computer dialogues and did not deal with any of the issues related to training novice users to use fairly complex human-computer interfaces. Two central training issues would be the use of embedded training and the use of generative computer-assisted instruction, both of which could become part of the operational dialogue interfaces. On a somewhat broader level, research is also needed to investigate the relationship among computer-aided training procedures, dialogue design, and the use of computer-based performance aiding as alternative mechanisms for optimizing human-computer interfaces.

Multi-operator interfaces. In many complex, computer-based systems, several operators are simultaneously coordinating their activities. Dialogue-design research activities need to be extended to multi-operator/computer interfaces in which human-to-human and shared human-computer dialogue interfaces are considered. Appropriate models for the design of those interfaces need to be developed.

Dynamic interfaces. Most of the human-computer interface transactions we have studied are static transactions that were completely specified at design time. For example, a menu on a graphic display followed by a user response is a typical transaction. However, there could be dynamic dialogue transactions modified by the interface user or the computational programs.

For example, the appearance of graphs or displays are not known until data are available. The difficulty in the design of dynamic interfaces is that it may be necessary for the dialogue author to specify the dynamics algorithmically, and this could add considerable complexity to the dialogue author's task. Dynamic interfaces need to be studied to determine whether it is, indeed, necessary to specify them algorithmically, as well as mechanisms for helping the dialogue author specify control structures and algorithmic constructs.

Implementation of human-computer guidelines. A fairly sizeable database of dialogue design considerations currently exists, and it will expand rapidly with the growing interest in human factors issues in the design of human-computer interfaces. Even if a completely comprehensive and nonconflicting database of design guidelines were available, it is critical to determine how this information can be best conveyed to the designer of dialogue software. The usual approach is merely to compile these considerations into a handbook with no retrieval assistance beyond a table of contents and/or index. Due to the complexity and overlapping nature of any comprehensive dialogue database, the organization of these handbooks and subsequent search for relevant guidelines quickly becomes unmanageable. Consequently, various forms of computer aiding should be considered. The computer aiding may be no more than tree searching procedures for data retrieval or may incorporate sophisticated rule-based procedures to aid in decision making.

OFFICE OF NAVAL RESEARCH

Engineering Psychology Group

TECHNICAL REPORTS DISTRIBUTION LIST

OSD

CAPT Paul R. Chatelier
Office of the Deputy under Secretary of Defense
OUSDRE (E&LS)
Pentagon, Room 3D129
Washington, D.C. 20301

Dr. Dennis Leedom
Office of the Deputy under Secretary of Defense (C^3I)
Pentagon
Washington, D.C. 20301

Department of the Navy

Engineering Psychology Group
Office of Naval Research
Code 442 EP
Arlington, VA 22217 (2 cys)

Dr. Andrew Rechnitzer
Office of the Chief of Naval Operations, OP952F
Naval Oceanography Division
Washington, D.C. 20350

Manpower, Personnel & Training Programs
Code 270
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217

Information Sciences Division
Code 433
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217

Special Assistant for Marine Corps Matters
Code 100M
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217

Department of the Navy

CDR James Offutt, Officer-in-Charge
ONR Detachment
1030 East Green Street
Pasadena, CA 91106

Director
Naval Research Laboratory
Technical Information Division
Code 2627
Washington, D.C. 20375

Dr. Michael Melich
Communications Sciences Division
Code 7500
Naval Research Laboratory
Washington, D.C. 20375

Dr. J.S. Lawson
Naval Electronic Systems Command
NELEX-06T
Washington, D.C. 20360

Dr. Neil McAlister
Office of Chief of Naval Operations
Command and Control
OP-094H
Washington, D.C. 20350

Dr. Robert G. Smith
Office of the Chief of Naval Operations, OP987H
Personnel Logistics Plans
Washington, D.C. 20350

Combat Control Systems Department
Code 35
Naval Underwater Systems Center
Newport, RI 02840

Human Factors Department
Code N-71
Naval Training Equipment Center
Orlando, FL 32813

Department of the Navy

Dr. Alfred F. Smode
Training Analysis and Evaluation Group
Naval Training & Equipment Center
Orlando, FL 32813

CDR Norman E. Lane
Code N-7A
Naval Training Equipment Center
Orlando, FL 32813

Dr. Gary Poock
Operations Research Department
Naval Postgraduate School
Monterey, CA 93940

Dean of Research Administration
Naval Postgraduate School
Monterey, CA 93940

Human Factors Engineering
Code 8231
Naval Ocean Systems Center
San Diego, CA 92152

Dr. A. L. Slafkosky
Scientific Advisor
Commandant of the Marine Corps
Code RD-1
Washington, D.C. 20380

Dr. L. Chmura
Naval Research Laboratory
Code 7592
Computer Sciences & Systems
Washington, D.C. 20375

Office of the Chief of Naval Operations (OP-115)
Washington, D.C. 20350

Professor Douglas E. Hunter
Defense Intelligence College
Washington, D.C. 20374

CDR C. Hutchins
Code 55
Naval Postgraduate School
Monterey, CA 93940

Human Factors Technology Administrator
Office of Naval Technology
Code MAT 0722
800 N. Quincy Street
Arlington, VA 22217

Department of the Navy

Commander
Naval Air Systems Command
Human Factors Programs
NAVAIR 334A
Washington, D.C. 20361

Commander
Naval Air Systems Command
Crew Station Design
NAVAIR 5313
Washington, D.C. 20361

Mr. Philip Andrews
Naval Sea Systems Command
NAVSEA 61R2
Washington, D.C. 20362

Commander
Naval Electronics Systems Command
Human Factors Engineering Branch
Code 81323
Washington, D.C. 20360

Larry Olmstead
Naval Surface Weapons Center
NSWC/DL
Code N-32
Dahlgren, VA 22448

Dr. George Moeller
Human Factors Engineering Branch
Submarine Medical Research Lab
Naval Submarine Base
Groton, CT 06340

Dr. Jerry Tobias
Auditory Research Branch
Submarine Medical Research Lab
Naval Submarine Base
Groton, CT 06340

Navy Personnel Research and Development Center
Planning & Appraisal Division
San Diego, CA 92152

Dr. Robert Blanchard
Navy Personnel Research and Development Center
Command and Support Systems
San Diego, CA 92152

Head, Aerospace Psychology Dept.
Code L5
Naval Aerospace Medical Research Lab
Pensacola, FL 32508

Department of the Navy

CDR J. Funaro
Human Factors Engineering Division
Naval Air Development Center
Warminster, PA 18974

Mr. Stephen Merriman
Human Factors Engineering Division
Naval Air Development Center
Warminster, PA 18974

Mr. Jeffrey Grossman
Human Factors Branch
Code 3152
Naval Weapons Center
China Lake, CA 93555

Human Factors Engineering Branch
Code 4023
Pacific Missile Test Center
Point Mugu, CA 93042

Dean of the Academic Departments
U.S. Naval Academy
Annapolis, MD 21402

Dr. W. Moroney
Human Factors Section
Systems Engineering Test
Directorate
U.S. Naval Air Test Center
Patuxent River, MD 20670

Human Factor Engineering Branch
Naval Ship Research and Development
Center, Annapolis Division
Annapolis, MD 21402

Department of the Army

Dr. Edgar M. Johnson
Technical Director
U.S. Army Research Institute
5001 Eisenhower Avenue
Alexandria, VA 22333

Technical Director
U.S. Army Human Engineering Labs
Aberdeen Proving Ground, MD 21005

Director, Organizations and
Systems Research Laboratory
U.S. Army Research Institute
5001 Eisenhower Avenue
Alexandria, VA 22333

Department of the Army

Mr. J. Barber
HQS, Department of the Army
DAPE-MBR
Washington, D.C. 20310

Department of the Air Force

U.S. Air Force Office of Scientific
Research
Life Sciences Directorate, NL
Bolling Air Force Base
Washington, D.C. 20332

AFHRL/LRS TDC
Attn: Susan Ewing
Wright-Patterson AFB, OH 45433

Chief, Systems Engineering Branch
Human Engineering Division
USAF AMRL/HES
Wright-Patterson AFB, OH 45433

Dr. Earl Alluisi
Chief Scientist
AFHRL/CCN
Brooks Air Force Base, TX 78235

Foreign Addresses

Dr. Kenneth Gardner
Applied Psychology Unit
Admiralty Marine Technology
Establishment
Teddington, Middlesex TW11 OLN
England

Director, Human Factors Wing
Defence & Civil Institute of
Environmental Medicine
P.O. Box 2000
Downsview, Ontario M3M 3B9
Canada

Dr. A.D. Baddeley
Director, Applied Psychology Unit
Medical Research Council
15 Chaucer Road
Cambridge, CB2 2EF England

Other Government Agencies

Defense Technical Information Center
Cameron Station, Bldg. 5
Alexandria, VA 22314 (12 copies)

Dr. Clinton Kelly
Defense Advanced Research Projects Agency
1400 Wilson Blvd.
Arlington, VA 22209

Dr. M. Montemerlo
Human Factors & Simulation Technology, RTE-6
NASA HQS
Washington, D.C. 20546

Other Organizations

Ms. Denise Benel
Essex Corporation
333 N. Fairfax Street
Alexandria, VA 22314

Dr. H. McI. Parsons
Essex Corporation
333 N. Fairfax
Alexandria, VA 22314

Dr. Jesse Orlansky
Institute for Defense Analyses
1801 N. Beauregard Street
Alexandria, VA 22311

Dr. J. O. Chinnis, Jr.
Decision Science Consortium
7700 Leesburg Pike
Suite 421
Falls Church, VA 22043

Dr. Paul E. Lehner
PAR Technology Corp.
P.O. Box 2005
Reston, VA 22090

Dr. Robert T. Hennessy
NAS - National Research Council (COHF)
2101 Constitution Avenue, N.W.
Washington, D.C. 20418

Dr. Deborah Boehm-Davis
General Electric Company
Information & Data Systems
1755 Jefferson Davis Highway
Arlington, VA 22202

Other Organizations

Mr. Edward M. Connelly
Performance Measurement Associates, Inc.
410 Pine Street, S.E.
Suite 300
Vienna, VA 22180

National Security Agency
ATTN: N-32, Marie Goldberg
9800 Savage Road
Ft. Meade, MD 20722

Mr. Joseph G. Wohl
Alphatech, Inc.
3 New England Executive Park
Burlington, MA 01803

Dr. Marvin Cohen
Decision Science Consortium, Inc.
Suite 721
7700 Leesburg Pike
Falls Church, VA 22043

Dr. Wayne Zachary
Analytics, Inc.
2500 Maryland Road
Willow Grove, PA 19090

Dr. William B. Rouse
School of Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, GA 30332

Dr. Richard Pew
Bolt Beranek & Newman, Inc.
50 Moulton Street
Cambridge, MA 02238

Dr. Douglas Towne
University of Southern California
Behavioral Technology Lab
3716 S. Hope Street
Los Angeles, CA 90007

Dr. David J. Getty
Bolt Beranek & Newman, Inc.
50 Moulton Street
Cambridge, MA 02238

Dr. Alan Morse
Intelligent Software Systems Inc.
Amherst Fields Research Park
529 Belchertown Rd.
Amherst, MA 01002